MINIMAC : a high speed access protocol based on a free access tree algorithm

Philippe Jacquet, Paul Mühlethaler

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 849

MINIMAC : A HIGH-SPEED ACCESS PROTOCOL BASED ON A FREE ACCESS TREE ALGORITHM

Philippe JACQUET
Paul MUHLETHALER

JUIN 1988

MINIMAC : A HIGH-SPEED ACCESS PROTOCOL BASED ON A FREE ACCESS TREE ALGORITHM

MINIMAC : UN PROTOCOLE D'ACCES HAUT-DEBIT BASE SUR UN PROTOCOLE EN ARBRE A ARRIVEE LIBRE

Philippe Jacquet and Paul Mühlethaler
INRIA, Rocquencourt
78153 LE CHESNAY CEDEX - FRANCE

ABSTRACT : This paper introduces a high-speed access protocol which is based on a free binary access tree algorithm. The maximum channel efficiency is 36 % and can reach 40% with ternary and quaternary tree algorithm. This protocol is very simple and very robust. It is a very good candidate to be a declaration protocol. In this paper we describe Minimac and analyze it in the case of an infinite population with a symmetric Poisson traffic.

RESUME : Ce papier introduit un protocole d'accès haut-débit qui est basé sur un algorithme en arbre binaire à arrivées libres. La capacité maximum est de 36 % et peut atteindre 40% avec un algorithme en arbre ternaire ou quaternaire. Ce protocole est très simple et très robuste. C'est un très bon candidat pour opérer comme protocole à déclaration. Dans ce papier nous décrivons Minimac et l'analysons dans le cas d'une population infinie avec un trafic de Poisson symétrique.

I. INTRODUCTION

When the propagation delay is not negligible compared with the transmission delay, we are under high-speed conditions. It may append with a high bandwidth or with long-haul communications. In the following, *a* denotes the ratio of the propagation delay and the transmission delay.

Existing protocols (e.g. Ethernet or the Token passing bus) are obviously not efficient for high-speed local area networks : too much time or bandwidth are lost in propagation delays. It can be shown that the maximum channel capacity C_{max} is for such protocols given by :

$$C_{max} = \frac{1}{1+0(a)} \text{ with } 0(a) \simeq a.$$

Expressnet has been one of the first to fit high-speed conditions [1] . With Lighnet [2] we have shown that it is possible to reach 100 % of channel efficiency and very good delays. Machnet has been one of the first design of high-speed protocol which can tolerate various kinds of topologies [3]. For example Machnet does not need stations physically ordered on the medium and therefore can be used for satellite communications. Machnet can also reach 100 % of channel efficiency.

As Machnet, Minimac can tolerate various kind of topologies but can only reach 36 % of channel efficiency. In spite of that, Minimac is very simple and robust, it is a very good candidate to be a declaration protocol [4].

This paper is divided into three parts. In the first part, we describe Minimac. In the second part, we analyze Minimac with standard hypothesis: symmetric Poisson traffic and infinite number of stations. In the third part, we compare results of the analytical model and those obtained out of simulations.

II. MINIMAC. A FREE ACCESS BINARY TREE ALGORITHM

First of all, Minimac is a synchron protocol. The time is divided in slots. The slot duration is taken to 1 and is supposed to fit the maximum packet length.

Minimac is a random access protocol, therefore collisions may occur. For a packet emitted on slot *0* the result : success or collision is known on the beginning of slot *b*. Quantity *b* includes the propagation delay and the time to detect the collision, therefore $b>a$. *b* must be the same for every stations. If it is not the case it can be easily done either physically or by a software delay.

To fill the slots between a transmission and the result of it (success or collision), we must allow the protocol to send other packets. We have in fact a weak parallelism where more than one packet can be waiting for a retransmission in a same station.

The access is free on every slot except for the stations which have to retransmit a packet.

Retransmission algorithm.

Minimac (contrary to Machnet) needs no permanent counters. Minimac has to initiate counters only when a collision is reported. If a collision is reported on slot *t* (created by the transmission of at least two packets on slot *t-b*) stations involved in this collision flip a 2 sided coin,the result is 1 or 0 with the same probability 0.5). In case of getting 1, the packet is immediatly retransmitted, otherwise a counter $S_i()$ is initiated to 1. At every slot $S_i()$ is updated, if the feedback is a collision $S_i()=S_i()+1$ otherwise $S_i()=S_i()$-1. Packet i is transmitted each slot t_i which satisfies $S_i(t_i)=0$ and the counter $S_i()$ is frozen when waiting for the feedback of its retransmission. After the successfull transmission of packet i, counter $S_i()$ is removed.

Minimac has only to deal with counters initialized when a collision is reported. The counter is removed when the corresponding packet is successfully transmitted. That clearly shows the simplicity of this protocol, its robustness and flexibility.
From a global point of view this protocol handles retransmissions with a virtual stack. If a collision is reported, packets getting 1 when flipping the coin are immediatly retransmitted as packets getting 0 are put on a new place at the top of the stack. A new place is created for them. When the feedback reports a success or an idle, packets at the top place of the stack are retransmitted and the place is removed. On every slot stations which have no packets to retransmit have free access.

III. ANALYTICAL MODEL

Our aim is to derive the distribution of the delay in the case of a symmetric Poisson traffic and with the assumption of an infinite population.

λ denotes the input load per slot (cumulated rate for all the nodes). C denotes the number of collisions and L the time spent by the packet in the stack. The waiting time for a packet is :

$$D = C \times (b + L).$$

We suppose that C and L are independent random variables. Introducing the generating functions of D, C and L one obtains :

$$D(u) = C(u^b \frac{1+L(u)}{2}),$$

where :

$$D(u) = \sum_{k \geq 0} D_k u^k,$$

$$C(u) = \sum_{k \geq 0} C_k u^k,$$

$$L(u) = \sum_{k \geq 0} L_k u^k.$$

where D_k denotes the probability a random packet has to wait k slots, C_k the probability a random packet experiences k collisions, L_k the probability that a random packet entered in the virtual stack remains k slots in it.
The following part is derived from [5] and is divided into two parts. In the first one we derived a formula for $C(u)$ and in the second one a formula for $L(u)$. We will just give here the most important formulas, the appendix contains all the intermediate calculations.

1.Computation of C(u).
Proposition: We have:

$$C(u) = \frac{W(\lambda,u)}{\lambda\,\phi(\lambda)},$$

where $\phi(\ )$ satisfies the functional equation:

$$\phi(u) = 1 + 2\phi(\frac{z}{2}+u) - e^{-z}\left(2\phi(\lambda)\,(1+z) + z\,\phi'(u)\right).$$

W(.,.) satisfies:

$$W(z,u) = 2\,W(\frac{z}{2}+\lambda,u) - \left(2\,W(\lambda,u)(1+z) + \frac{\partial W}{\partial z}(\lambda,u)\right) z\,e^{-z}$$

$$+z\left(\omega(z,u) - \omega(\frac{z}{2}+\lambda,u) - (1-\omega(z,u))e^{-z}\right), \qquad A$$

with $w(z, u)$ satisfying:

$$\omega(z,u) = u\,\omega(\frac{z}{2}+\lambda,u) + (1-u\,\omega(\lambda,u))e^{-z}. \qquad B$$

Demonstration:

Let $\omega_n(u)$ be the generating function for the number of collisions experienced by a packet if its first collision involves n-1 other packets.

$$\omega_n(u) = \sum_{k\geq 0} \omega_n^{\,k} u^k,$$

$$\omega_1(u) = 1.$$

ω_n^k is the probability that a packet experiences k collisions if its first collision involves n-1 other packets

The tree algorithm is "recursive" in the sense that two packets which did not collide during the first transmission of one of them will never collide during their retransmissions. Let us suppose that we have an initial collision of multiplicity n. According to the recursivity of the algorithm, it is possible to track and isolate the slots dedicated for the resolution of this specific collision. Let us call this set of slots a n session. By forgetting the delay between transmissionand feedback this n session is equivalent to a n session with $b=0$. This case refers to [5] and the following analysis, focused on evaluations about n sessions, makes a constant use of techniques involved in [5].

If we introduce :

$$\omega(z,u) = \sum_{n\geq 0} \omega_{n+1}(u) \frac{z^n}{n!} e^{-z},$$

it can be shown via algebric manipulations of generating functions (Appendix) that $\omega(z,u)$ satisfies the functionnal equation :

$$\omega(z,u) = u\,\omega(\frac{z}{2} + \lambda, u) + (1 - u\,\omega(\lambda, u))e^{-z}.$$

The idea is then to define a random generating function P_n(u) for a session started with a collision involving n packets, the coefficient of u^j is the number of packets that experienced j collisions before being transmitted. Letting $W_n(u)$ be the expected value of P_n(u) we can define :

$$W(z,u) = \sum_{n\geq 0} W_n(u) \frac{z^n}{n!} e^{-z}.$$

$W_n(z,u)$ satisfies the functional equation (Appendix):

$$W(z,u) = 2\,W(\frac{z}{2} + \lambda, u) - \left(2\,W(\lambda, u)(1+z) + \frac{\partial W}{\partial z}(\lambda, u)\right) z\, e^{-z}$$

$$+ z\left(\omega(z,u) - \omega(\frac{z}{2} + \lambda, u) + (1 - \omega(z,u))e^{-z}\right). \qquad B$$

$W(z,u)$ denotes the generating function of the number of collisions for packets belonging to the same session. As $\phi(\lambda)$ gives the mean length of a session, $\lambda\phi(\lambda)$ gives the mean number of packets in a session therefore:

$$C(u) = \frac{W(\lambda,u)}{\lambda\,\phi(\lambda)},$$

End of demonstration. The appendix includes algebric manipulations which lead to equations A and B and a reminder on how solve these equations.

2. Calculation of $L(u)$.
Theorem 2:

$$L(u) = \frac{1 - \sqrt{1 - 4\alpha\beta u^2}}{2\alpha u},$$

with α ,β verifying

$$\alpha = \frac{\phi(\lambda) - 1}{2\phi(\lambda)} \; and \; \alpha + \beta = 1.$$

Demonstration:
The incursion in stack is supposed to be a random walk. The level in the stack L increases of 1 if there is a collision (probability α) and discreases of 1 otherwise (probability β, $\alpha + \beta = 1$). Stability implies $\alpha < \beta$.

One defines :

$$L_n(u) = \sum_{b \geq 0} L_n^k u^k ,$$

where L_n^k denotes the probability that if the level in the stack is n at slot 0 then k is the first slot when this level reaches 0.

Obviously we have :

$$L_n(u) = (L_1(u))^n .$$

It comes :

$$L_n(u) = u(\alpha L_{n+1}(u) + \beta L_{n-1}(u)),$$

$$L_1(u) = \alpha u L_1^2(u) + \beta u .$$

Thus :

$$L_1(u) = \frac{1 - \sqrt{1 - 4\alpha\beta u^2}}{2\alpha u} .$$

In a session the number of collisions is the number of non collision minus 1, therefore the average number of collision in a session is :

$$\frac{\phi(\lambda) - 1}{2} .$$

It comes :

$$\alpha = \frac{\phi(\lambda) - 1}{2\phi(\lambda)} \; and \, L(u) = L_1(u) = \frac{1 - \sqrt{1 - 4\alpha\beta u^2}}{2\alpha u} .$$

End of demonstration.

Corollary: The mean acces delay can be directly derived, by resolving a specific functional equation. We have $E[W] = D'(1)$ and by denoting $\Omega(z) = \partial W / \partial u(z,1)$ we obtain:

$$E[W] = \frac{\Omega(\lambda)}{\lambda\phi(\lambda)} \left(b + \frac{\phi(\lambda)}{2} \right).$$

The generating function $\Omega(z)$ satisfies the following functional equation

$$\Omega(z) = 2\Omega\left(\frac{z}{2} + \lambda\right) + z - \left(2\Omega(z)(1+z) + \Omega'(z)z\right)e^{-z} .$$

The solution can be easily computed by methods recalled in the appendix. The variance can also be derived by the same technique.

IV. COMPARISON BETWEEN SIMULATIONS AND THE ANALYTICAL MODEL.

Simulation software has been written in C^{++} and uses Sphinx [6] an event driven simulator. Numerical computations use also C^{++} and its capability to define operators on vectors. In the simulations we have set to 100 the number of connected stations. The traffic is Poisson and symmetric. We first focuse on the mean time between the first try to transmit a packet and its successfull transmission. Figure 3 and 4 give the comparison between results obtained out of simulations and with the analytical model. We have investigated the case of $b=10$ and $b=20$. The matching is good, simulations and the analytical model provide results whithin a few percents except close to the saturation. We have also compared the probability distribution function derived from the results of simulations and from the analytical model. The matching is still very good except close to the saturation, see figure 5,6 and 7.

V FURTHER RESULTS.

We have analysed the free access binary tree algorithm for high speed communications. The Q-ary tree algorithms [7] are alternative to binary tree algorithms. When a collision occurs, conflicting stations are split in Q subpopulations instead of 2. From the station point of view, when a collision is reported the involved stations flip a Q-sided coin, the result is 0,1,....,Q-1 with probability 1/Q and is reported as the new value of $S_i(\ )$. When $S_i(.)>0$, at every slot, if the feedback is a collision, then $S_i(.)=S_i(.)+Q-1$, else $S_i(.)=S_i(.)-1$.
The following table shows the maximum utilization λ_{max}(in percent of the channel bandwidth) that are respectively entailed by the Q-ary trees (from[7])

Q	λ_{max}	v_{crit}
2	36,0177	0,5
3	40,1599	0,333
4	39,9293	0,25
5	38,7241	0,2
6	37,3354	0,1667
7	35,9731	0,1428

The optimal tree algorithm is $Q=3$. The Q-ary tree algorithms adapted to high speed condition are tractable to analysis with the same techniques as for the binary one [7]. For example, the generating function $\phi(z)$ of the length of the session satisfies the following functional equation

$$\phi(z)=Q\phi(\frac{z}{Q}+\lambda)+1-\left(Q\,\phi(\lambda)(1+z)+\phi'(\lambda)z\right)e^{-z}.$$

The generating function related to the number of collision satisfies:

$$\Omega(z)=Q\,\Omega\left(\frac{z}{Q}+\lambda\right)+z-\left(Q\,\Omega(\lambda)(1+z)+\Omega'(\lambda)z\right)e^{-z}.$$

The resolution of this functional equation leads in particular to the mean access delay

$$E[W]=\frac{\Omega(\lambda)}{\lambda\phi(\lambda)}\left(b+\frac{Q-1}{2}\phi(\lambda)\right)$$

which is illustrated by figure 8. This figure clearly shows that the greater Q is the lesser are the delays, but the lesser is the maximum admissible throughput. It seems that the optimal tree algorithm may be $Q=4$.
This last assertion should be tempered by the fact that the binary shows the highest robustnest to channel error. Let us suppose that every slot is misinterpreted as collision with probability v. It is interesting to determine v_{crit} above which the algorithm is unstable whatever be λ [8]. It is easy to see that the Q- ary algorithms entail $v_{crit}=1/Q$. The table below also shows the v_{crit}'s. One figure 9 one can find the variance of the delay for $Q=2$ computed with the analytical model.

VI CONCLUSIONS

We have proposed a robust and flexible algorithm to solve the problem of medium access control for high speed communications. Our algorithm is more efficient for delays and more simple than the one proposed by Capetenakis with several trees working in parallel every propagation delay. This protocol is also very fault-tolerant. We have shown also that it is possible to find a fairly accurate analytical model for it. Matching between simulations and the analytical model are good.
Our protocol can be used with a Q-ary tree algorithm; the bigger Q is , the better are the delays, but the lesser is the maximum admissible throughput.

References

[1] F Tobagi, F Borgonovo, L Fratta, "Expressnet: a high-performance integrated-services local area network" IEEE Journal on selected areas in communications,vol.sac-1,No 5, November 1983, pp. 898-913 .

[2] Philippe Jacquet, Paul Mühlethaler. Rapport de recherche INRIA N°654. Lightnet: Un protocole d'accès à haut débit pour bus unidirectionnel. Avril 1987

[3] Philippe Jacquet, Paul Mühlethaler. Rapport de recherche INRIA N°684. Machnet: Un réseau performant pour le haut-débit ou les communications longues distances. Juin 1987

[4] Philippe Jacquet, Paul Mühlethaler.Frequency allocation for high speed communications. HSLAN 88 Liège April 1988 pp 8.01- 8.09.

[5] G Fayolle, P Flajolet, M.Hofri, P Jacquet. "Analysis of a stack algorithm for randoom multi-access communication" IEEE trans on inf theory, Vol 31 , No 2, pp 244-255, March 1985.

[6] T Billoir, P Sabarthes, "A highly flexible performance evaluation tool for local area networks" Advanced Seminar on Real-Time Local Area Networks .Bandol, France organized by INRIA and sponsored by the US Army European Research Office. April 16-18, 1986 pp. 207-223.

[7]P Mathys, P Flajolet. " Q ary collision resolution algorithms in randoom access systems with free or blocked channel access"IEEE trans on inf theory, Vol 31 , No 2, pp 217-243, march 1985.
[8]J Massey " Collision resolution algorithm and random access algorithms" in multi-user Comuniaction Systems, G Longo, Ed ,CISM Courses and lecture Notes 265 New York: Spring Verlag, 1981, pp 73-137.
[9]G Fayolle, P Flajolet M.Hofri "On a functional equation arising in the analysis of a protocol for multi-access broadcast channel". Advances in Applied Probability, 18, pp 441-472 1986.

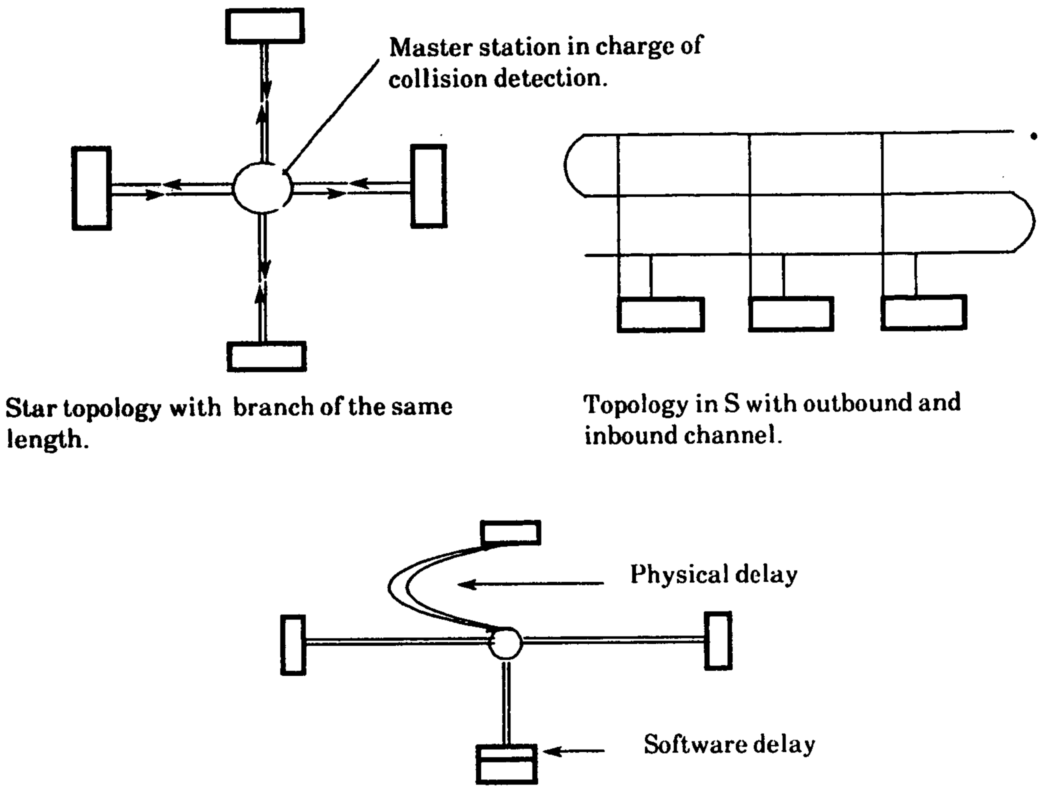

FIG 1 Various kind of topologies for Minimac

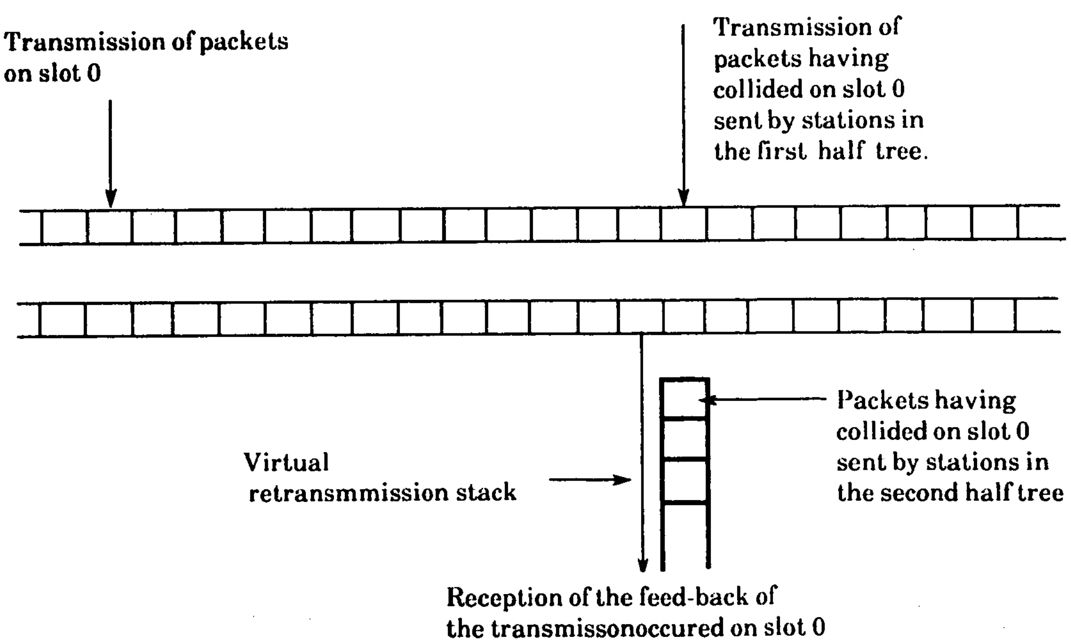

FIG 2 Retransmission algorithm with a virtual stack.

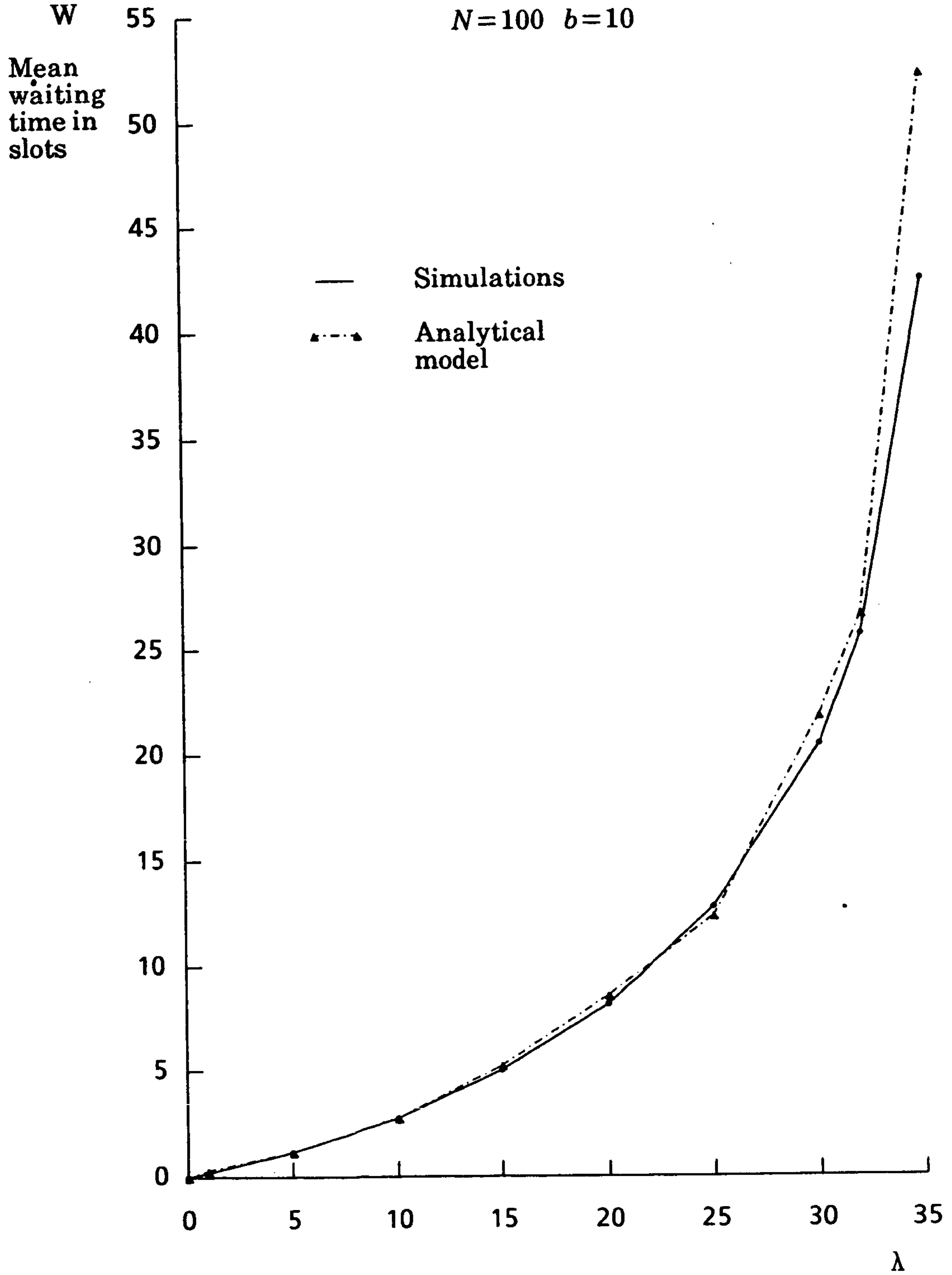

FIG 3. Mean access delay W versus input load λ in percent of channel capacity.

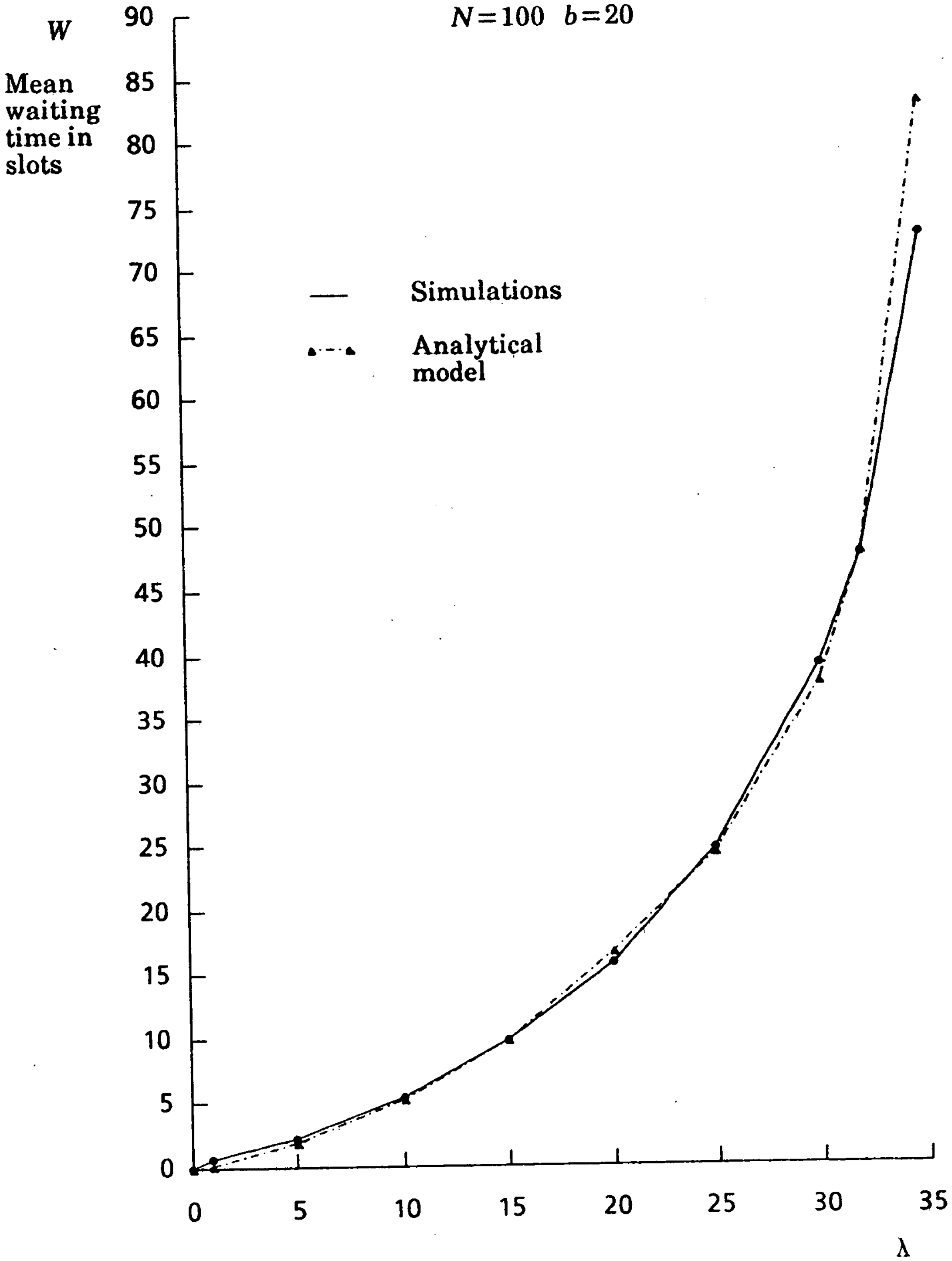

FIG 4. Mean access delay W versus input load λ in percent of channel capacity.

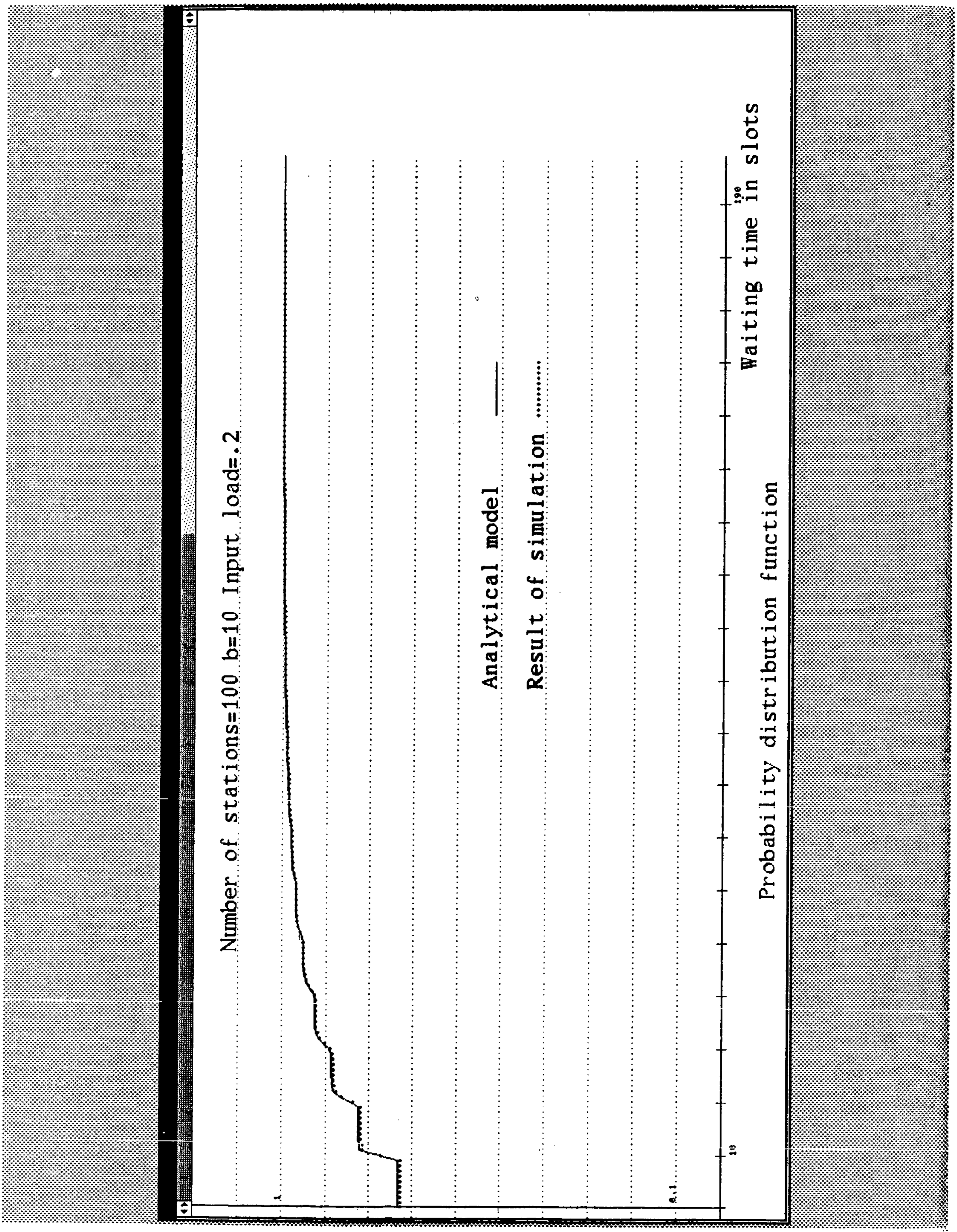
Number of stations=100 b=10 Input load=.2
Analytical model
Result of simulation
Probability distribution function
Waiting time in slots

FIG 5.

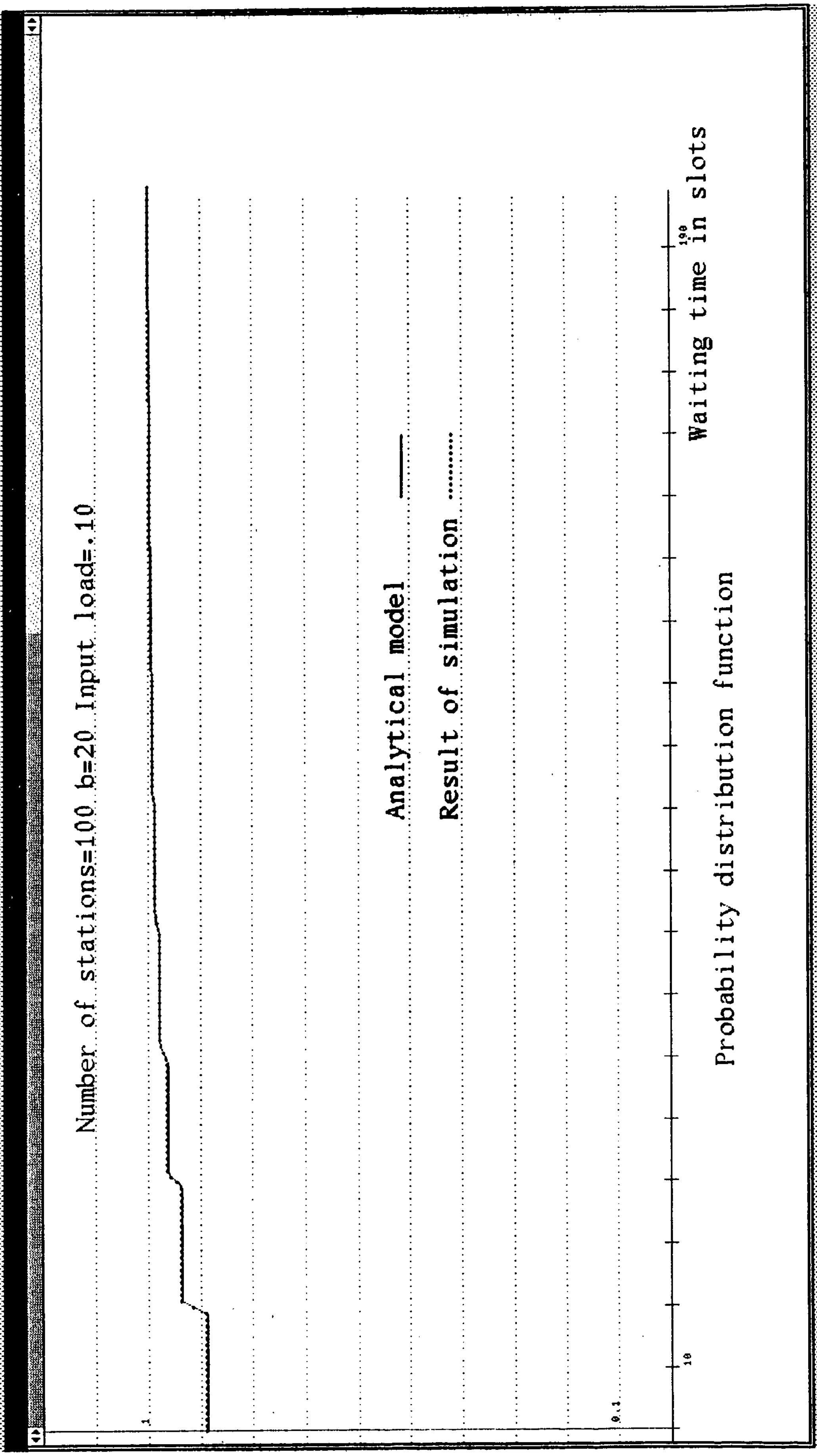
Number of stations=100 b=20 Input load=.10
Analytical model
Result of simulation
Probability distribution function
Waiting time in slots
1
0.1
10
100

FIG 6.

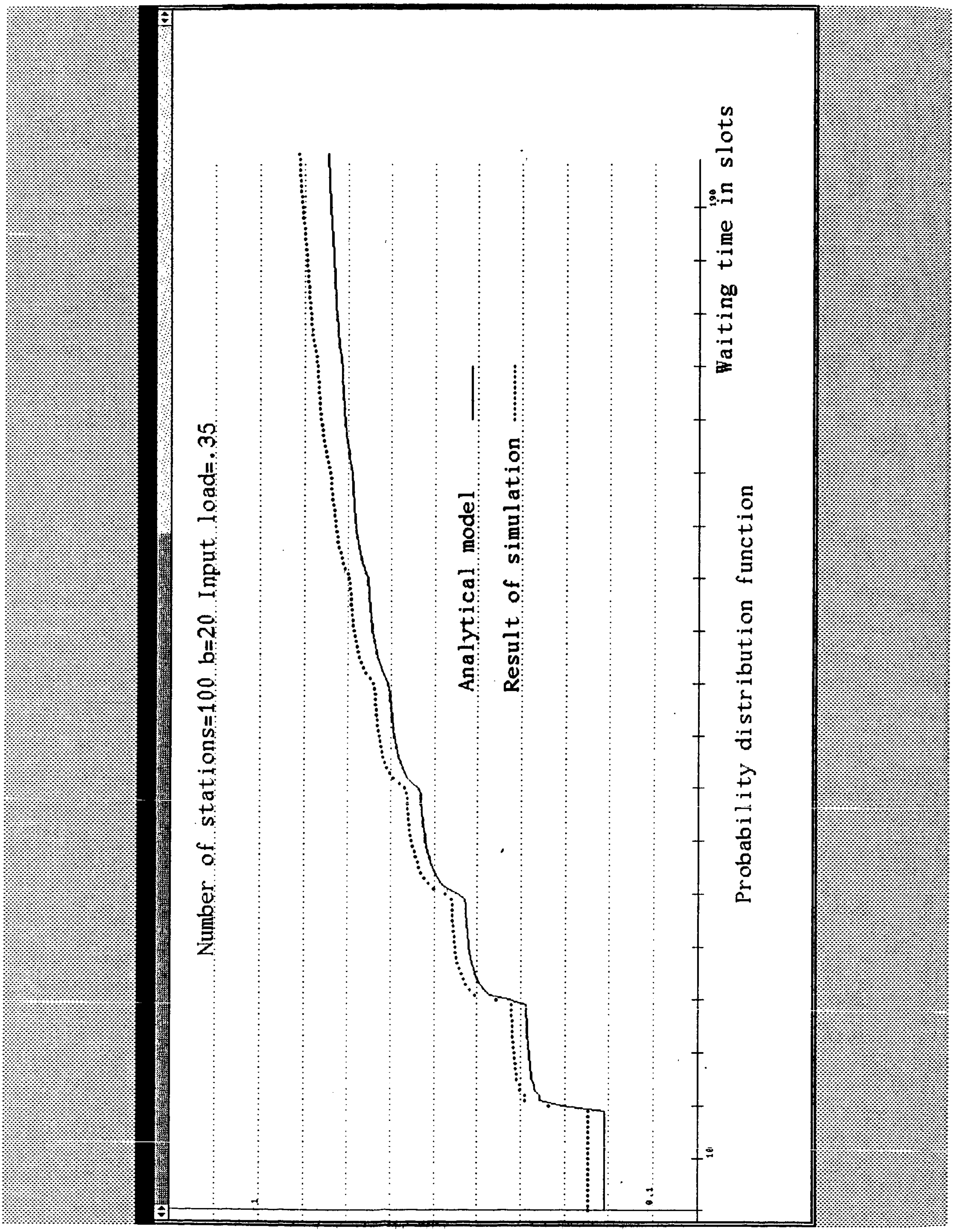
Number of stations=100 b=20 Input load=.35
Analytical model
Result of simulation
Waiting time in slots
Probability distribution function

FIG 7.

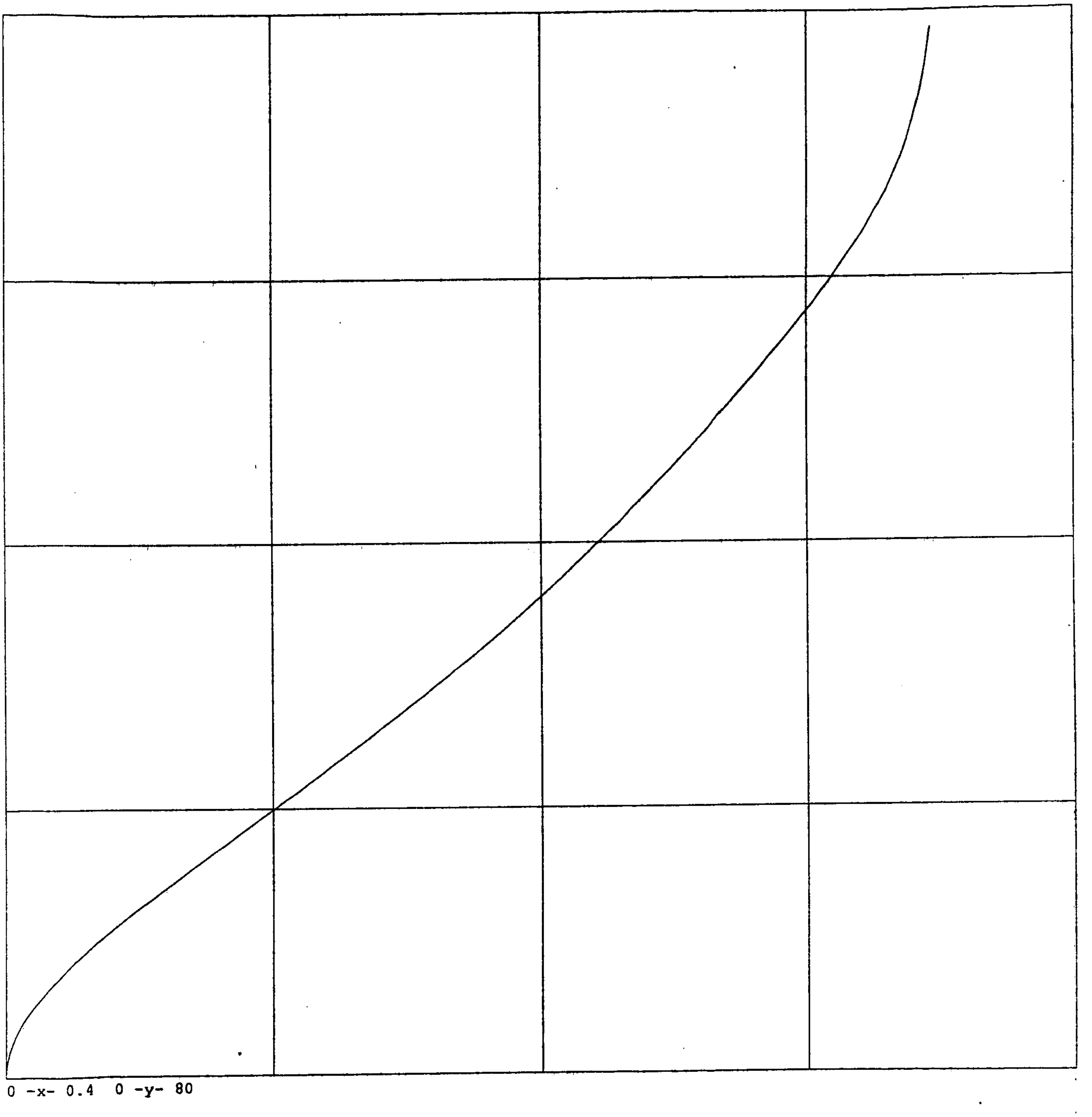

FIG 8. Mean access delay for different values of Q.

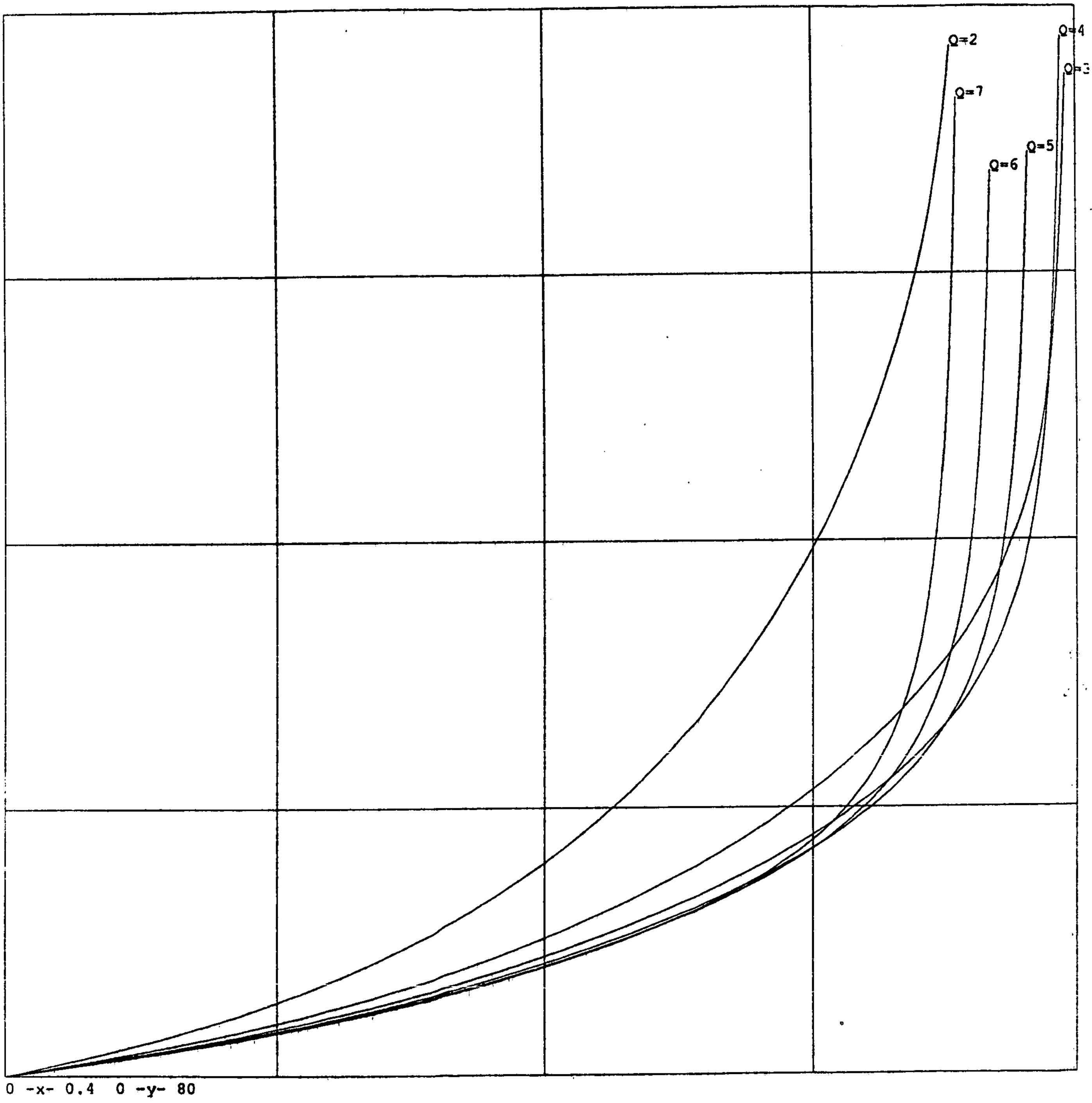

FIG 9. Variance of the delay for Q = 2.

Appendix

This appendix contains four parts. The first two parts are dedicated to derivations of equations A and B as in the last two ones we give the solutions of these equations.

a.Derivation of equation B

Let $X_n(u)$ be the random function of u related to any session defined by an initial collision of multiplicity n :

$$X_n(u) = \frac{1}{n} \sum_{\textit{initial packets of a n session}} u^{\textit{number of collisions of the packet}} .$$

The coefficient of u^j is the number of packets initially colliding in a given n session that experienced j collisions before being transmitted, divided by n : we have:

$$X_1(u) = 1$$

and, for $n \geq 2$.

$$n X_n(u) = (n-j)\, u X_{n-j+x}(u) + j\, u X_{j+y}(u).$$

where j has a binomial distribution Prob$(j=p) = 2^{-n}/(p!\,(n-p)!)$ $p \in \{0,1,..,n\}$, x and y are Poisson processes with rate λ.

Clearly we have :

$$\omega_n(u) = E[X_n(u)]$$

and thus :

$$\omega_n(u) = u \sum_{y \geq 0} e^{-\lambda} \frac{\lambda^y}{y!} \sum_{x \geq 0} e^{-\lambda} \frac{\lambda^x}{x!} \sum_{j=0}^{n} \frac{2^{-n}}{j!(n-j)!} \left(\frac{n-j}{n} w_{n-j+x}(u) + \frac{j}{n} w_{j+y}(u) \right)$$

therefore

$$\omega_n(u) = u \sum_{j=0}^{n} \frac{2^{-n}}{j!(n-j)!} \left(\frac{j}{n} \sum_{x \geq 0} e^{-\lambda} \frac{\lambda^x}{x!} w_{j+x}(u) + \frac{n-j}{n} \sum_{y \geq 0} e^{-\lambda} \frac{\lambda^y}{y!} w_{n-j+y}(u) \right)$$

$$\omega_n(u)\frac{z^{n-1}}{(n-1)!}e^{-z}=e^{-z}u\sum_{j=1}^{n}\frac{z^{j-1}}{2^n(j-1)!}\frac{z^{n-j}}{(n-j)!}\sum_{x\geq 0}e^{-\lambda}\frac{\lambda^x}{x!}\omega_{j+x}(u)$$

$$+e^{-z}u\sum_{j=0}^{n-1}\frac{z^j}{2^n j!}\frac{z^{n-1-j}}{(n-j-1)!}\sum_{x\geq 0}e^{-\lambda}\frac{\lambda^x}{x!}\omega_{n-j+x}(u)$$

$$\sum_{n\geq 1}w_n(u)\frac{z^{n-1}}{(n-1)!}e^{-z}=$$

$$e^{-z}u\frac{1}{2}e^{-\lambda}\sum_{n\geq j}\frac{z^{n-j}}{2^{n-j}(n-j)!}\sum_{k\geq 1}\omega_k(u)\sum_{j=1}^{k}\frac{z^{j-1}}{2^{-j+1}(j-1)!}\frac{\lambda^{k-j}}{(k-j)!}$$

$$+e^{-z}u\frac{1}{2}e^{-\lambda}\left(\sum_{n\geq j}\frac{z^{n-j}}{2^{n-j}(n-j)!}\sum_{k\geq 1}\omega_k(u)\sum_{j=1}^{k}\frac{z^{j-1}}{2^{-j+1}(j-1)!}\frac{\lambda^{k-j}}{(k-j)!}\right)$$

$$-e^{-z}\left(u\frac{1}{2}\sum_{x\geq 0}\omega_{1+x}(u)\frac{\lambda^x}{x!}e^{-\lambda}+u\frac{1}{2}\sum_{x\geq 0}\omega_{1+x}(u)\frac{\lambda^x}{x!}e^{-\lambda}\right)+e^{-z}$$

$$=\frac{1}{2}u\,e^{-z}e^{z/2}\sum_{k\geq 1}\omega_k(u)\frac{(\frac{z}{2}+\lambda)^{k-1}}{(k-1)!}+\frac{1}{2}u\,e^{-z}e^{z/2}\sum_{k\geq 1}\omega_k(u)\frac{(\frac{z}{2}+\lambda)^{k-1}}{(k-1)!}$$

$$+e^{-z}-u\,e^{-z}(\frac{1}{2}+\frac{1}{2})\omega(\lambda,u).$$

That leads to the anounced equation :

$$\omega(z,u)=u\,\omega(\frac{z}{2}+\lambda,u)+(1-u\,\omega(\lambda,u))e^{-z}.$$

β)Derivation of equation A

We are now looking for the functionnal equation satisfied by $W(z,u)$. $W_n(u)$ satisfies the following equation (see [5]).

$$W_n(u)=\sum_{y\geq 0}e^{-\lambda}\frac{\lambda^y}{y!}\sum_{x\geq 0}e^{-\lambda}\frac{\lambda^x}{x!}\sum_{j=0}^{n}\frac{2^{-j}}{j!}\frac{2^{-n+j}}{(n-j)!}\Big(W_{j+x}(u)+W_{n-j+y}(u)$$

$$+n\,\omega_n(u)-j\,\omega_{j+x}(u)-(n-j)\,\omega_{n-j+y}(u)\Big)$$

With the same algebric manipulations as for equation B it comes :

$$W(z,u) = 2\,W(\frac{z}{2}+\lambda,u) - (2\,W(\lambda,u)(1+z) + \frac{\partial W}{\partial z}(\lambda,u))e^{-z}$$

$$+\, z\,(\omega(z,u) - \omega(\frac{z}{2}+\lambda,u) - (1-\omega(\lambda,u))e^{-z}).$$

Our aim now is to eliminate ∂W/∂Z (z,u) to find the following equation:

$$W(z,u) = 2W(\frac{z}{2}+\lambda,u) - 2W(\lambda,u)(1+\frac{z}{1-2\lambda})e^{-z} - K(\lambda,u)\,z\,e^{-z}$$

$$+\, z\left(\omega(z,u) - \omega\,(\frac{z}{2}+\lambda,u) - (1-\omega(z,u))\,e^{-z}\right),$$

where :

$$K(\lambda,u) = \frac{e^{2\lambda}}{1-2\lambda}\left(\lambda\,\frac{\partial\omega}{\partial z}(2\lambda,u) + (2\lambda-1)(1-\omega(\lambda,u)e^{-2\lambda}\right)$$

If we derive the functional equation satisfied by $W(.,.)$,we get :

$$\frac{\partial W}{\partial z}(z,u) = \frac{\partial\omega}{\partial z}(\frac{z}{2}+\lambda,u) + \left(2\,W(\lambda,u)(1+z) + \frac{\partial W}{\partial z}(\lambda,u)z\right)e^{-z}$$

$$-\left(2\,W(\lambda,u) + \frac{\partial W}{\partial z}(\lambda,u)z\right)e^{-z} + \omega(z,u) - \omega(\frac{z}{2}+\lambda,u) - \left(1-\omega(\lambda,u)\right)e^{-z}$$

$$+\, z\,\frac{\partial W}{\partial z}(z,u) - \frac{z}{2}\,\omega(\frac{z}{2}+\lambda,u) + z\,e^{-z}(1-\omega(\lambda,u))$$

If $z = 2\lambda$, it comes successively:

$$\frac{\partial W}{\partial z}(\lambda,u)e^{-2\lambda}(1-2\lambda) = 4\lambda\,W(\lambda,u)e^{-2\lambda} + (2\lambda-1)\left(1-\omega(\lambda,u)\right)e^{-2\lambda} + \lambda\,\omega'(2\lambda,u)$$

$$\frac{\partial W}{\partial z}(\lambda,u) = 4\lambda\,W(\lambda,u)\,\frac{1}{1-2\lambda} + \frac{e^{2\lambda}}{1-2\lambda}\,\lambda\,\frac{\partial W}{\partial z}(2\lambda,u) + \left(1-\omega(\lambda,u)\right)^{\omega}(2\lambda,u)$$

$$\frac{\partial W}{\partial z}(\lambda,u) = \frac{4\lambda\,W(\lambda,u)}{1-2\lambda} + K(\lambda,u)$$

That leads to the announced formula for $W(z,u)$.

ξ)Resolution of equation B :

Following methods can be found in [9]

Let :

$$\sigma(z) = \frac{z}{2} + \lambda ,$$

$$\sigma^n(z) = \frac{z}{2^n} + (2 - \frac{1}{2^n}) \lambda .$$

If we derive (B) we obtain :

$$\frac{\partial \omega}{\partial z}(z,u) = \frac{u}{2} \frac{\partial \omega}{\partial z}(\sigma(z),u) - (1 - u\, \omega(\lambda,u))e^{-z} .$$

By iteration we have :

$$\frac{\partial \omega}{\partial \omega}(z, u) = -(1 - u\, \omega(\lambda ,u)) \sum_{n \geq 0} \frac{u^n}{2^n} e^{-\sigma^n(z)} .$$

It can be integrated term by term. It comes :

$$\omega(z,u) = (1 - u\, \omega(\lambda , u)) \sum_{n \geq 0} u^n \, [e^{-\sigma^n(0)} - e^{-\sigma^n(z)}] .$$

Introducing :

$$F(z,u) = \sum_{n \geq 0} u^n (e^{-\sigma^n(0)} - e^{-\sigma^n(z)}),$$

we have :

$$\omega(\lambda, u) = \frac{1 - F(\lambda,u)}{1 - uF(\lambda,u)} ,$$

and :

$$\omega(z,u) = 1 - \frac{1 - u}{1 - u F(\lambda,u)} F(z,u),$$

8)Resolution of equation A:

If we omit u, equation A has the following type [9]:

$$\phi(z) = 2\phi(\frac{z}{2} + \lambda) + h(z),$$

Deriving with regard to z we have :

$$\phi'(z) = \phi'(\frac{z}{2} + \lambda) + h'(z),$$

$$\phi''(z) = \frac{1}{2} \phi''(\frac{z}{2} + \lambda) + h''(z).$$

The last equation leads to :

$$\phi''(z) = \sum_{n\geq 0} (\frac{1}{2})^n h''(\sigma^n(z)) \ .$$

If we integrate it term by term, we obtain successively :

$$\phi'(z) - \phi'(0) = \sum_{n\geq 0} h'(\sigma^n(z)) \ - h''(\sigma^n(0),$$

$$\phi(z) - z\,\phi'(0) - \phi(0) = \sum_{n\geq 0} 2^n(h(\sigma^n(z)) - h(\sigma^n(0) - \frac{z}{2^n} h'(\sigma^n(0))).$$

Let us introduce the following notation:

$$S[h(z),t] = \sum_{n=0}^{\infty} 2^n \left(h(\sigma^n(t)) - h(\sigma^n(0)) - \frac{t}{2^n}(h(\sigma^n(t)) \right) .$$

Therefore :

$$W(z,u) = z - 2W(\lambda,u) S[(1+\frac{z}{1-2\lambda})e^{-z}] - K(\lambda,u) S[ze^{-z},z]$$

$$+ S[z(\omega(z,u) - \omega(\frac{z}{2}+\lambda,u) - (1-\omega(\lambda,u))e^{-z}), z].$$

$$W(\lambda,u) = \frac{1}{1+2S[(1+\frac{z}{1-2\lambda})e^{-z},\lambda]} \left(\lambda - K(\lambda,u) S[ze^{-z},\lambda] \right.$$

$$\left. + S[z(\omega(z,u) - e^{-z} - \frac{1}{u}(\omega(z,u) - e^{-z})),\lambda] \right).$$

In [5] it can be found that :

$$\phi(\lambda) = \frac{1}{1+2S[(1+\frac{z}{1-2\lambda})e^{-z},\lambda]} ,$$

therefore :

$$C(u) = \frac{1}{\lambda}\left(\lambda - K(\lambda,u) S[ze^{-z},\lambda] + S[z(\omega(z,u) - e^{-z} - \frac{1}{u}(\omega(z,u) - e^{-z}),\lambda] \right). \qquad \square$$

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique